

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-167447

(43)Date of publication of application : 24.06.1997

(51)Int.Cl.

G11B 20/12

(21)Application number : 07-330059

(71)Applicant : SONY CORP

(22)Date of filing : 19.12.1995

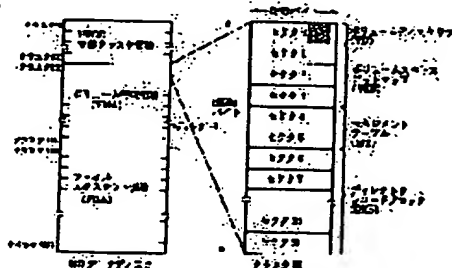
(72)Inventor : SUZUKI YUICHI

(54) DEVICE AND METHOD FOR RECORDING/REPRODUCING DATA AND DATA RECORDING MEDIUM

(57)Abstract:

PROBLEM TO BE SOLVED: To quickly determine whether file management information is correct or not.

SOLUTION: A fault detect byte(FDB) is provided in the free byte of a volume descriptor(VD) in a volume management area(VMA). When data is recorded in a disk, this FDB is set to 0×42 . After the writing of the data is finished, its value is rewritten to be 0. When a system-down occurs in a data recording/ reproducing device in the middle of data writing, the fault detect byte(FDB) is kept at 0×42 . On the other hand, when no system-down occurs the value of the fault detect byte(FDB) is kept at 0. Thus, the existence of the abnormality of data management information is determined based on the value of the fault detect byte(FDB).



LEGAL STATUS

[Date of request for examination] 04.07.2001

[Date of sending the examiner's decision of rejection] 27.09.2004

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3641863

[Date of registration] 04.02.2005

[Number of appeal against examiner's decision of rejection] 2004-22074

[Date of requesting appeal against examiner's decision of rejection] 27.10.2004

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平9-167447

(43)公開日 平成9年(1997)6月24日

(51)IntCl. ³	識別記号	庁内整理番号	FI	技術表示箇所
G11B 20/12	102	9295-5D	G11B 20/12	102

審査請求 未請求 請求項の数9 OL (全13頁)

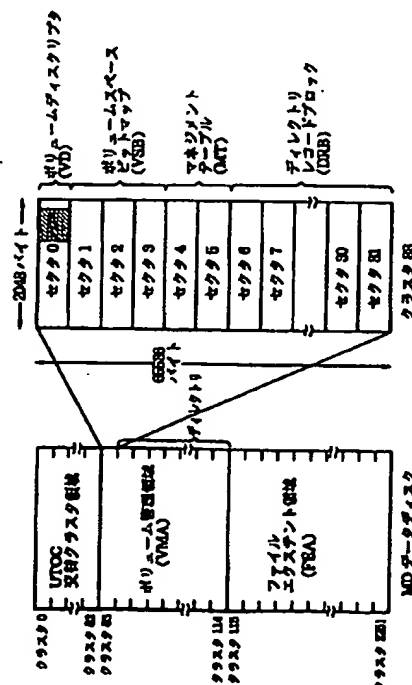
(21)出願番号	特願平7-330059	(71)出願人	000002185 ソニー株式会社 東京都品川区北品川6丁目7番35号
(22)出願日	平成7年(1995)12月19日	(72)発明者	鈴木 雄一 東京都品川区北品川6丁目7番35号 ソニー株式会社内
		(74)代理人	弁理士 稲本 義雄

(54)【発明の名称】 データ記録再生装置および方法並びにデータ記録媒体

(57)【要約】

【課題】 ファイル管理情報が正しいか否かを迅速に判定できるようにする。

【解決手段】 ボリューム管理領域(VMA)中のボリュームディスクリプタ(VD)の空きバイトに、異常検出バイト(FDB)を設ける。ディスクにデータを記録するとき、このFDBを0x42に設定する。そして、データの書き込みが終了した後、その値を0に書き換える。データ記録再生装置が、データの書き込み途中においてシステムダウンしたとき、異常検出バイト(FDB)は、0x42のままとなる。これに対して、途中でシステムダウンが発生しなければ、異常検出バイト(FDB)の値は0のままとなる。従って、異常検出バイト(FDB)の値から、データの管理情報の異常の有無を判定することができる。



【特許請求の範囲】

【請求項1】 記録媒体にデータ記録する前に、前記データを記録する前の状態であることを表す第1の識別符号を記録する第1の記録手段と、

前記第1の識別符号の記録の後に、前記データを前記記録媒体に記録する第2の記録手段と、

前記第2の記録手段による前記データの記録の後に、前記第1の識別符号を、前記データを記録した後の状態であることを表す第2の識別符号に書き換える第3の記録手段とを備えることを特徴とするデータ記録再生装置。

【請求項2】 前記第1の記録手段は、前記データ記録再生装置の使用が開始されたとき、前記第1の識別符号を記録することを特徴とする請求項1に記載のデータ記録再生装置。

【請求項3】 前記記録媒体から、前記第1または第2の識別符号を再生する再生手段と、

前記再生手段により再生された前記第1または第2の識別符号に対応して、前記記録媒体の修復処理を行う修復手段とをさらに備えることを特徴とする請求項1に記載のデータ記録再生装置。

【請求項4】 前記第3の記録手段は、前記データ記録再生装置の動作終了が指令されたとき、前記第2の識別符号を記録することを特徴とする請求項1に記載のデータ記録再生装置。

【請求項5】 前記第1の記録手段と第3の記録手段は、前記第1の識別符号と前記第2の識別符号を、前記記録媒体に記録することを特徴とする請求項1に記載のデータ記録再生装置。

【請求項6】 前記記録媒体に対して前記データを記録することを表す第3の識別符号を記録する第4の記録手段をさらに備え、

前記第1の記録手段は、前記記録媒体に対して前記データを実際に記録するとき、前記第1の識別符号を記録することを特徴とする請求項1に記載のデータ記録再生装置。

【請求項7】 前記第1の記録手段と第3の記録手段は、前記第1の識別符号と第2の識別符号を、複数回、前記記録媒体に記録することを特徴とする請求項1に記載のデータ記録再生装置。

【請求項8】 記録媒体にデータ記録する前に、前記データを記録する前の状態であることを表す第1の識別符号を記録し、

前記第1の識別符号の記録の後に、前記データを前記記録媒体に記録し、

前記データの記録の後に、前記第1の識別符号を、前記データを記録した後の状態であることを表す第2の識別符号に書き換えることを特徴とするデータ記録方法。

【請求項9】 データを記録するデータ記録媒体において、前記データとともに、前記データを記録する前の状態、

または前記データを記録した後の状態であることを表す識別符号が記録されていることを特徴とするデータ記録媒体。

【発明の詳細な説明】**【0001】**

【発明の属する技術分野】 本発明はデータ記録再生装置および方法並びにデータ記録媒体に関し、特に、データ記録媒体中に記録されているデータを管理する管理情報が正しいか否かを迅速に識別することができるようにした、データ記録再生装置および方法並びにデータ記録媒体に関する。

【0002】

【従来の技術】 最近、コンピュータの普及にともない、磁気ディスク、光磁気ディスク、ICメモ리카ードなどのデータ記録媒体に種々のデータが記録され、利用されるようになってきた。

【0003】 図15は、光磁気ディスクの一種であるMD（ミニディスク）（商標）データのファイルシステムの構成例を表している。同図に示すように、MDデータディスクは、2252個のクラスタに区分され、クラスタ0乃至クラスタ82の83個の領域は、UTOCおよび交替クラスタ領域とされ、クラスタ83乃至クラスタ114は、ボリューム管理領域（Volume Management Area（VMA））とされている。また、クラスタ115乃至クラスタ2251は、データを記録するファイルエクステンツ領域（File Extent Area（FEA））とされている。

【0004】 UTOCおよび交替クラスタ領域中のクラスタ3乃至5は、UTOC（User Table of Content）領域とされ、クラスタ50乃至82は、交替クラスタ領域とされている。その他のクラスタは、リザーブとされている。

【0005】 ボリューム管理領域（VMA）の先頭のクラスタ83には、その先頭のセクタに、ボリュームディスクリプタ（Volume Descriptor（VD））が配置され、次の3セクタには、ボリュームスペースビットマップ（Volume Space Bitmap（VSB））が配置され、続く2セクタには、マネージメントテーブル（Management Table（MT））が配置され、さらに続く26セクタには、ディレクトリレコードブロック（Directory Record Block（DRB））が配置されている。

【0006】 1クラスタは32セクタにより構成され、1セクタは2048バイトとされている。従って、1クラスタは65536バイト（64kバイト）となる。

【0007】 ボリュームディスクリプタ（VD）は、図16に示すように、このディスクに記録されているボリュームスペースビットマップ（VSB）、マネージメントテーブル（MT）、ディレクトリレコードブロック（DRB）、ファイルエクステンツ領域（FEA）の位置を表している。

【0008】ボリュームスペースビットマップ (VSB) は、ファイルエクステント領域 (FEA) の空き領域を管理するビットマップデータを有している。マネージメントテーブル (MT) は、ディレクトリレコードブロック (DRB) の空きを管理している。また、ディレクトリレコードブロック (DRB) は、ファイルとディレクトリの管理を行う領域とされる。1つのDRBは、1つのセクタに記述されている。

【0009】図17は、ボリューム管理領域 (VMA) のフォーマットを模式的に表している。同図に示すように、クラスタ83の最初のセクタにはボリュームディスクリプタ (VD) が、次の3セクタにはボリュームスペースビットマップ (VSB) が、次の2セクタには、マネージメントテーブル (MT) が、それぞれ配置されている。そして、クラスタ83の、それ以降の各セクタと、クラスタ84乃至クラスタ114の各セクタには、ディレクトリレコードブロック (DRB) 6乃至DRB 1023が配置されている。

【0010】図18は、従来のMDデータ記録再生装置における処理例を表している。

【0011】最初にステップS91において、ディスク (MDデータ) から、ボリュームディスクリプタ (VD)、ボリュームスペースビットマップ (VSB) およびマネージメントテーブル (MT) が読み取られ、データ記録再生装置のRAM (図示せず) に書き込まれる。次にステップS92において、記録、再生、消去のいずれかの指令が入力されるまで待機する。

【0012】記録、再生または消去の指令が入力されたとき、それぞれステップS93、S94またはS95に進み、指令に対応する記録処理、再生処理、または消去処理が実行される。すなわち、ディスクのファイルエクステント領域 (FEA) に対して、データが記録され、再生され、あるいは消去される。

【0013】次にステップS96において電源オフの指令、またはディスクを排出する指令などが入力されるなどして、動作の終了が指示されたか否かが判定される。終了が指示されない場合においては、ステップS92に戻り、それ以降の処理を繰り返し実行する。

【0014】ステップS96において終了が指令されたと判定された場合、ステップS97に進み、RAMに記憶されているボリュームディスクリプタ (VD)、ボリュームスペースビットマップ (VSB)、マネージメントテーブル (MT) の内容が更新されている場合、それがディスクに書き込まれる。

【0015】このようにして、ファイルエクステント領域 (FEA) にデータを記録したり、記録されているデータを消去したりした場合においては、それに対応して、ボリューム管理領域 (VMA) の内容を更新することで、ボリューム管理領域 (VMA) の内容を参照して、ファイルエクステント領域 (FEA) に記録されて

いる任意のファイルを自由にアクセスすることができるようになされている。

【0016】しかしながら、このようにボリューム管理領域 (VMA) の内容を全てRAMに記憶しておき、動作を終了させるとき、最後にまとめてその内容をディスクに書き込むようにすると、RAMの容量を大きくしなければならず、コスト高となるばかりでなく、終了を指令した後、実際に動作が終了されるまでの時間が長くなり、操作性が悪化する。

【0017】そこで、ボリューム管理領域 (VMA) の内容を変更するたびに、その内容をディスクに記録するようにすることも考えられる。しかしながらそのようにすると、使用中における処理速度が低下することになる。

【0018】これを解決するために、ボリュームディスクリプタ (VD)、ボリュームスペースビットマップ (VSB)、およびマネージメントテーブル (MT) をRAMに常駐させ、ディレクトリレコードブロック (DRB) は、アクセス頻度の高いものをRAM上に記憶させるようにし、新しいディレクトリレコードブロック (DRB) をアクセスするときは、古くなったディレクトリレコードブロック (DRB) をディスク上に書き込み、新しいディレクトリレコードブロック (DRB) をRAMに取り込むようにするLeast Recently Used (LRU) アルゴリズムが用いられる場合がある。

【0019】しかしながら、そのようにすると、RAM上には、ディスクにまだ書き込まれていない情報が残ることになり、しかも、使用頻度の高い情報ほど、RAMに常駐される結果、ディスクには書き込まれていない可能性が高くなる。

【0020】その結果、装置の故障、停電、操作ミスなどにより、上述したファイルシステムの動作手順が完結しない状態で、データ記録再生装置が停止してしまう (システムダウンする) 状態が発生すると、システムエラーが発生する。

【0021】例えば、ディレクトリレコードブロック (DRB) 32に所定のファイルAが登録され、その内容データもファイルエクステント領域 (FEA) に書かれたタイミングで、システムダウンが発生したとすると、ボリュームスペースビットマップ (VSB) がまだ更新されていないので、以後のアクセス時に、ファイルエクステント領域 (FEA) のファイルAが記録されている領域は、空き領域と見なされてしまう。その結果、次に新しいファイルBを記録しようとする、ファイルAが記録されている領域にファイルBが上書きされ、ファイルAが消去されてしまう事態が発生する。

【0022】また、例えば、親ディレクトリのディレクトリレコードブロック (DRB) 6において、サブディレクトリCが、ディレクトリレコードブロック (DRB) 64として登録されており、ディレクトリレコード

ブロック (DRB) 64 が、まだ更新されていない状態でシステムダウンすると、実際には、サブディレクトリ C は存在しないことになる。その結果、サブディレクトリ C を読み込むと、システムエラーとなってしまふ。

【0023】

【発明が解決しようとする課題】このように、ディスクのファイル管理情報に異常があると、ファイル (データ) を正確に保持することが困難になる。そこで、ディスクのファイル管理情報に異常があるか否かを検査し、異常が発見された場合には、これを修復するようにしている。

【0024】ディスクのファイル管理情報に異常があるか否かの検査は、例えば次のような項目を検査することにより行われる。

【0025】(1) 各ディレクトリに登録されているサブディレクトリが、実際に存在するか否か。

【0026】(2) 各ディレクトリに登録されている各ファイルの記録領域に対して、ボリュームスペースビットマップ (VSB) が使用中となっているか否か。

【0027】(3) その他、ファイル管理情報の自己矛盾がないか否か。

【0028】これらの検査には、数分の時間が必要となる。そこで、例えばデータ記録再生装置の電源がオンされた場合、あるいは、データ記録再生装置にディスクが装着されたとき、この検査を行うようにすることができる。しかしながらそのようにすると、データ記録再生装置を実際に使用し始めるまでに時間がかかりすぎ、操作性が損なわれることになる。

【0029】そこで、装置の動作に異常が認められたとき、検査するようにすることができる。しかしながらそのようにすると、大切なデータが破壊された後、異常が発見されることになり、信頼性が損なわれることになる。

【0030】本発明はこのような状況に鑑みてなされたものであり、迅速かつ確実に、記録媒体上のデータ管理情報が正しいか否かを検出することができるようにするものである。

【0031】

【課題を解決するための手段】請求項1に記載のデータ記録再生装置は、記録媒体にデータ記録する前に、データを記録する前の状態であることを表す第1の識別符号を記録する第1の記録手段と、第1の識別符号の記録の後に、データを記録媒体に記録する第2の記録手段と、第2の記録手段によるデータの記録の後に、第1の識別符号を、データを記録した後の状態であることを表す第2の識別符号に書き換える第3の記録手段とを備えることを特徴とする。

【0032】請求項8に記載のデータ記録方法は、記録媒体にデータ記録する前に、データを記録する前の状態であることを表す第1の識別符号を記録し、第1の識別

符号の記録の後に、データを記録媒体に記録し、データの記録の後に、第1の識別符号を、データを記録した後の状態であることを表す第2の識別符号に書き換えることを特徴とする。

【0033】請求項9に記載のデータ記録媒体は、データとともに、データを記録する前の状態、またはデータを記録した後の状態であることを表す識別符号が記録されていることを特徴とする。

【0034】請求項1に記載のデータ記録再生装置においては、第1の記録手段が、記録媒体にデータ記録する前に、データを記録する前の状態であることを表す第1の識別符号を記録し、第2の記録手段が、第1の識別符号の記録の後に、データを記録媒体に記録し、第3の記録手段が、第2の記録手段によるデータの記録の後に、第1の識別符号を、データを記録した後の状態であることを表す第2の識別符号に書き換える。

【0035】請求項8に記載のデータ記録方法においては、記録媒体にデータ記録する前に、データを記録する前の状態であることを表す第1の識別符号を記録し、第1の識別符号の記録の後に、データを記録媒体に記録し、データの記録の後に、第1の識別符号を、データを記録した後の状態であることを表す第2の識別符号に書き換える。

【0036】請求項9に記載のデータ記録媒体においては、データとともに、データを記録する前の状態、またはデータを記録した後の状態であることを表す識別符号が記録される。

【0037】

【発明の実施の形態】図1は、本発明のデータ記録再生装置の構成例を示すブロック図である。データ記録媒体1は、例えば、磁気ディスク、光磁気ディスク、ICメモ리카ードなどよりなる記録媒体であって、データが記録される。媒体駆動部2は、データ記録媒体1に対してデータを記録し、また、これを再生するものであり、例えばデータ記録媒体1が光磁気ディスクである場合には、これを回転するスピンドルモータ、光磁気ディスクに対してレーザ光を照射する光ピックアップ、各種のサーボ動作を行うサーボ回路、記録信号を変調し、また、再生信号を復調する変復調回路などよりなるディスクドライブ装置により構成される。

【0038】RAM3は、データ記録媒体1に対してデータを記録再生する場合におけるバッファとしての機能や、CPU5が各種の動作を実行する上において、必要なデータやプログラムなどを適宜記憶するワークエリアとしても機能する。不揮発性メモリ4は、データ記録再生装置の電源をオフした後も記憶しておく必要のあるデータを記憶する。CPU5は、ROM6に記憶されているプログラムに従って、各種の処理を実行する。

【0039】データ入力部7は、各種のデータを入力するものであり、キーボード、デジタルインタフェース、

A/Dコンバータなどにより構成される。データ出力部8は、データを出力する部分であり、例えばデジタルインタフェース、CRTディスプレイ、D/Aコンバータなどにより構成される。

【0040】以下の実施例においては、データ記録媒体1はMDデータであるとする。

【0041】図2は、データ記録媒体1としてのMDデータのフォーマットを表している。このフォーマットは、図15を参照して説明した従来のMDデータのフォーマットと基本的に同様であるが、この実施例においては、ボリューム管理領域(VMA)のクラスタ83に配置されているボリュームディスクリプタ(VD)の未使用バイトに、異常検出バイト(Fault Detect Byte(FDB))が配置される。その他の構成は、図15における場合と同様である。

【0042】この異常検出バイト(FDB)は、その値が、0ならば正常、0x42ならば異常と定義される。なお、ここで0xは16進数を表す。

【0043】この異常検出バイト(FDB)は、図3のフローチャートに示すように記録される。すなわち、いま、ユーザがデータ入力部7を操作して、データの記録を指令すると、ステップS1(第1の記録手段)において、CPU5は、媒体駆動部2を制御し、データ記録媒体1としてのMDデータのボリュームディスクリプタ(VD)中の異常検出バイト(FDB)に0x42(第1の識別符号)を書き込ませる。

【0044】次に、ステップS2(第2の記録手段)に進み、CPU5は、データ入力部7より入力された記録データをRAM3に一時的に記憶させる。そして、RAM3に記憶されたデータを読み出し、これを媒体駆動部2を介して、データ記録媒体1に供給し、ファイルエクステンション領域(FEA)に記録させる。このデータの記録にともなう、ボリューム管理領域(VMA)の管理情報も更新される。

【0045】そして、ステップS3(第3の記録手段)に進み、CPU5は、ステップS1で0x42を書き込んだディスク(MDデータ)中の異常検出バイト(FDB)を0(第2の識別符号)に書き換える。

【0046】このように、この実施例によれば、MDデータにデータを書き込む前に、異常検出バイト(FDB)に0x42を書き込み、データの書き込み完了後、その値を0に書き換えるようにしているので、データの書き換え処理の途中でシステムダウンが発生した場合、MDデータの異常検出バイト(FDB)は、0x42のままとなる。

【0047】これに対して、データの書き込み前か、書き換え完了後にシステムダウンが発生したとしても、ディスクの内容は正常である。そしてこのとき、異常検出バイト(FDB)の値は0となっている。

【0048】そこで、データ記録再生装置にディスクを

装着したとき、あるいはデータ記録再生装置の電源がオンされたとき、MDデータの異常検出バイト(FDB)を読み出し、その値が0であれば、そのままその後の処理を継続し、その値が0x42である場合には、ディスクの内容を修復する修復プログラムを実行するようにする。

【0049】しかしながら、図3の実施例によると、ディスク上のデータを書き換えるたびに、異常検出バイト(FDB)を書き換える処理(ステップS1とS3の処理)を行わなければならない、処理速度が低下する。そこで、例えば図4のフローチャートに示すように処理することができる。

【0050】図4の処理においては、最初にステップS11(再生手段)において、CPU5は、データ記録媒体1(MDデータ)からボリュームディスクリプタ(VD)、ボリュームスペースビットマップ(VSB)およびマネージメントテーブル(MT)を読み取り、それをRAM3に記憶させる。次にステップS12において、いま読み取ったボリュームディスクリプタ(VD)中の異常検出バイト(FDB)を読み取り、その値が0x42であるか否かを判定する。

【0051】後述するように、この実施例においても、データの記録動作の途中でシステムダウンが発生したとき、異常検出バイト(FDB)は、0x42となっている。そこで、この場合、データの内容が正常ではないと判定し、ステップS13(修復手段)に進み、修復プログラムを実行する。これにより、データ記録媒体1中の異常が修復される。すなわち、ディレクトリに登録されているサブディレクトリが実際に存在するようになされ、各ディレクトリに登録されている各ファイルの記録領域に対し、ボリュームスペースビットマップ(VSB)が使用中となり、その他、管理情報の自己矛盾がないように修復が行われる。

【0052】このような修復が行われたので、次にステップS14に進み、RAM3上のボリュームディスクリプタ(VD)中の異常検出バイト(FDB)の値を0に書き換える。そして、ステップS15において、RAM3に記憶されているボリュームディスクリプタ(VD)、ボリュームスペースビットマップ(VSB)およびマネージメントテーブル(MT)をデータ記録媒体1に書き込ませる(MDデータの内容を更新する)。これにより、MDデータのボリュームディスクリプタ(VD)には、異常検出バイト(FDB)の値として0が記録されることになる。

【0053】そこで、ステップS11に戻り、再びMDデータからボリュームディスクリプタ(VD)、ボリュームスペースビットマップ(VSB)およびマネージメントテーブル(MT)を読み取り、RAM3に書き込むと、そのRAM3に書き込まれた異常検出バイト(FDB)の値は0となっている。そこで、この場合において

は、ステップS12において、NOの判定が行われ、ステップS16に進む。

【0054】ステップS16においては、RAM3に記憶されている異常検出バイト(FDB)の値を0から0x42に書き換えさせる。そして、ステップS17において、RAM3に記憶されているボリュームディスクリプタ(VD)、ボリュームスペースビットマップ(VSB)およびマネージメントテーブル(MT)が読み出され、MDデータに書き込まれる。

【0055】この図4の処理は、データ記録再生装置にデータ記録媒体1を装着したとき、あるいは、データ記録媒体1が、既にデータ記録再生装置に装着されている場合においては、データ記録再生装置の電源をオンしたときに開始される。すなわち、データ記録再生装置の使用を開始するとき実行される。

【0056】従って、データ記録媒体1に対する実質的な使用を開始する前に、データ記録媒体1が、常に正常な状態とされる。また、この状態において、データ記録媒体1の異常検出バイト(FDB)は、その後、途中でシステムダウンが発生した場合に備えて、その値が0x42に設定される。

【0057】次に、ステップS18において、CPU5は、データ入力部7から所定の動作の指令が入力されるまで待機し、記録が指令されたとき、ステップS19に進み記録処理を実行し、再生が指令されたとき、ステップS20に進み再生処理を実行し、消去が指令されたとき、ステップS21に進み消去処理を実行する。

【0058】図5は、記録処理の詳細を表している。最初にステップS111において、CPU5は、RAM3に記憶されているボリュームスペースビットマップ(VSB)を走査し、データを記録する上において必要な数の空きセクタを検索する。そしてステップS112に進み、ステップS111で検索した空きセクタにデータを記録する。

【0059】なお、MDデータの場合、記録はクラスタ単位で行われる。従って、データを新たに記録すべき空きセクタが属するクラスタの他のセクタに、既にデータが記録されている場合においては、そのセクタのデータが読み出され、RAM3に記憶される。そして、データ入力部7より入力された記録データがRAM3のそのクラスタの空きセクタ中に書き込まれる。そして、1クラスタ分のデータが集められた後、その1クラスタ分のデータがRAM3から読み出され、媒体駆動部2を介してMDデータの対応するファイルエクステンション領域(FEA)のクラスタに書き込まれる。

【0060】次にステップS113に進み、CPU5は、RAM3に記憶されているボリュームスペースビットマップ(VSB)の、いまデータを記録した空きセクタに対応するビットを使用済に設定する。

【0061】次にステップS114に進み、いま、デー

タを記録したセクタに対応するディレクトリレコードブロック(DRB)をデータ記録媒体1から読み出し、RAM3に記憶する。そして、ステップS115において、いま読み込んだディレクトリレコードブロック(DRB)に、ステップS112で記録されたデータにより構成されるファイルのファイル名、データを記録した最初のセクタの番号、データを記録したセクタの数などを記録し、さらにこれをMDデータに記録する。

【0062】図6は、再生処理の詳細を表している。再生処理においては、ステップS121において、CPU5は、データ記録媒体1に記録されているディレクトリレコードブロック(DRB)をRAM3に読み出して走査し、データ入力部7から入力された再生すべきファイル名を検索する。目的のファイル名のディレクトリレコードブロック(DRB)が検索されたとき、ステップS122に進み、そのディレクトリレコードブロック(DRB)からそのファイルの開始セクタ番号やセクタ数を読み取る。そしてステップS123において、CPU5は、ステップS122で読み取った開始セクタ番号から、やはり読み取った数のセクタを再生する。

【0063】この再生データはRAM3に記憶された後、再び読み出され、データ出力部8に出力される。

【0064】図7は、消去処理の詳細を表している。消去処理においては、ステップS141において、データ記録媒体1のディレクトリレコードブロック(DRB)がRAMに読み出されて走査され、消去対象とされるファイル名を有するディレクトリレコードブロック(DRB)が検索される。目的のファイル名を有するディレクトリレコードブロック(DRB)が検索されたとき、ステップS142に進み、そのディレクトリレコードブロック(DRB)の開始セクタ番号とセクタ数が読み取られる。

【0065】次にステップS143に進み、CPU5は、ステップS142で読み取られた開始セクタ番号のセクタから、そのファイルを構成する数のセクタのボリュームスペースビットマップ(VSB)上の対応するビットをクリアする。このボリュームスペースビットマップ(VSB)は、RAM3に記憶されているものである。そして、ステップS144に進み、データ記録媒体1中のディレクトリレコードブロック(DRB)における、そのファイルの記述を消去する。

【0066】図4に戻って、以上のようにして、ステップS19乃至S21において、記録処理、再生処理、または消去処理が実行された後、ステップS22に進み、処理の終了が指令されたか否かを判定する。データ入力部7よりデータ記録媒体1の排出や、データ記録再生装置の電源をオフする指令などが入力されるなどして、終了が指令されていないければ、ステップS18に戻り、それ以降の処理を繰り返し実行する。

【0067】処理の終了が指令されたと判定された場

合、ステップS23に進み、CPU5は、RAM3の異常検出バイト(FDB)を0に書き換える。この異常検出バイト(FDB)は、処理開始時において、ステップS16で、0x42に書き換えられていたものである。そして、ステップS24に進み、RAM3に記憶されているボリュームディスクリプタ(VD)、ボリュームスペースビットマップ(VSB)およびマネジメントテーブル(MT)をデータ記録媒体1に書き込む。さらに、ステップS25において、終了指令に対応して、電源をオフしたり、ディスクを排出する処理を実行する。

【0068】以上のように、図4の実施例によれば、ステップS18乃至S22の実質的な使用状態の前と後に、それぞれ1回ずつ異常検出バイト(FDB)を書き換える処理を実行するだけなので、処理速度の低下は抑制される。

【0069】しかしながら、図4に示す実施例の場合、ディスクにデータを書き込む場合だけでなく、単に再生を行う場合にも、異常検出バイト(FDB)の書き込み処理が実行されることになる。そこで、データ記録媒体1に対してデータの記録動作を行わない再生処理だけが行われる場合においては、データ記録媒体1の異常検出バイト(FDB)の更新を行わないようにすることも可能である。図8は、この場合の実施例を表している。

【0070】図8においては、最初にステップS41において、データ記録媒体1からボリュームディスクリプタ(VD)、ボリュームスペースビットマップ(VSB)およびマネジメントテーブル(MT)が読み取られ、RAM3に記憶される。そして、ステップS42において、いま、読み取られたボリュームディスクリプタ(VD)中の異常検出バイト(FDB)が0x42であるか否かが判定される。異常検出バイト(FDB)が0x42である場合においては、ステップS43に進み、修復プログラムが実行される。そして、修復処理が実行された後、ステップS44において、RAM3の異常検出バイト(FDB)が0に書き換えられ、RAM3に記憶されているボリュームディスクリプタ(VD)、ボリュームスペースビットマップ(VSB)およびマネジメントテーブル(MT)が、ステップS45において、データ記録媒体1に書き込まれる。以上の処理は、図4におけるステップS11乃至S15の処理と同様の処理である。

【0071】ステップS42において、RAM3に読み取った異常検出バイト(FDB)が0x42ではないと判定された場合(0であると判定された場合)、ステップS46(第4の記録手段)に進み、CPU5は、RAM3の異常検出バイト(FDB)の値を0から0x42に書き換える。また、RAM3(または不揮発性メモリ4)に記憶されている検査バイト(FDS)(第3の識別符号)を0に設定する。この検査バイト(FDS)は、データ記録媒体1に対してデータを書き込んだか否

かを表す識別符号として機能する。ここでは、データ記録媒体1にデータが書き込まれないものとして、値0を初期設定する。

【0072】次に、ステップS47に進み、データ入力部7より所定の指令が入力されるまで待機する。

【0073】データの記録が指令されたとき、ステップS48に進み、CPU5は、RAM3(また不揮発性メモリ4)に記憶されている検査バイト(FDS)が0であるか否かを判定する。まだ一度もデータ記録媒体1に対してデータを書き込んでいない場合においては、上述したように、ステップS46で検査バイト(FDS)に0が初期設定されている。この場合においては、ステップS49に進み、RAM3に記憶されているボリュームディスクリプタ(VD)、ボリュームスペースビットマップ(VSB)およびマネジメントテーブル(MT)がデータ記録媒体1に書き込まれる。従って、データ記録媒体1の異常検出バイト(FDB)は、0x42とされる。

【0074】次にステップS50に進み、CPU5は、RAM3に記憶されている検査バイト(FDS)の値を1に書き換える。これにより、データ記録媒体1に対して記録処理が行われたこと(続くステップS51で行われる)が設定される。そしてステップS51に進み、データ入力部7より入力され、RAM3に記憶したデータを、データ記録媒体1のファイルエクステンション領域(FEA)に記録させる。このステップS51における記録処理の詳細は、図5に示した場合と同様である。

【0075】ステップS48において、検査バイト(FDS)が0ではない(1である)と判定された場合、データ記録媒体1に対して、1回以上データの書き込み処理が実行されたことになる。この場合、既にデータ記録媒体1の異常検出バイト(FDB)は、0x42に既に書き換えられているので、ステップS49、S50の処理はスキップされ、ステップS51に進み、記録処理が直ちに開始される。

【0076】一方、ステップS47において、データの消去が指令された場合、この場合においても、データ記録媒体1の書き込みが行われるため、記録における場合と同様の処理が実行される。すなわち、この場合においては、ステップS53において、RAM3に記憶されている検査バイト(FDS)が0であるか否かが判定され、0であればステップS54に進み、RAM3に記憶されているボリュームディスクリプタ(VD)、ボリュームスペースビットマップ(VSB)およびマネジメントテーブル(MT)が、データ記録媒体1に書き込まれる。そして、ステップS55において、RAM3に記憶されている検査バイト(FDS)が1に設定される。その後、ステップS56に進み、消去処理が実行される。この消去処理の詳細は、図7に示した場合と同様である。

【0077】ステップS53において、検査バイト(FDS)が0ではない(1である)と判定された場合、既にデータ記録媒体1には、1回以上データの書き込み処理が行われていることになる。そこで、この場合においては、異常検出バイト(FDB)は、 0×42 の値に既に書き換えられている。このため、ステップS54、S55の処理はスキップされ、ステップS56の消去処理が直ちに実行される。

【0078】さらに、ステップS47において、再生の指令が入力されたとき、ステップS52に進み、再生処理が実行される。この再生処理の詳細は、図6に示した場合と同様である。

【0079】すなわち、再生が指令された場合においては、記録または消去が指令された場合とは異なり、データ記録媒体1の異常検出バイト(FDB)を更新する処理は行われないことになる。

【0080】ステップS51、S52またはS56の次に、ステップS57に進み、処理の終了が指令されたか否かが判定され、指令されていなければステップS47に戻り、それ以降の処理が繰り返し実行される。

【0081】処理の終了が指令されたと判定された場合、ステップS58に進み、検査バイト(FDS)が1であるか否かが判定される。上述したように、データ記録媒体1に対して1回でもデータを記録したとき、ステップS50またはS55において、検査バイト(FDS)には1が設定され、ステップS49またはS54において、データ記録媒体1中の異常検出バイト(FDB)が 0×42 の値に書き換えられている。そこで、ステップS59において、RAM3のボリュームディスクリプタ(VD)中の異常検出バイト(FDB)を0に書き換える。そして、ステップS60において、RAM3に記憶されているボリュームディスクリプタ(VD)、ボリュームスペースビットマップ(VSB)およびマネージメントテーブル(MT)をデータ記録媒体1に書き込む処理を実行する。

【0082】さらに、ステップS61に進み、終了の指令に対応して、電源オフまたはディスクを排出する処理が実行される。

【0083】ステップS58において、検査バイト(FDS)が1ではない(0である)と判定された場合、データ記録媒体1には、再生処理が行われたとしても、データの書き込み処理(記録または消去処理)が一度も行われていないことになる。従って、データ記録媒体1の異常検出バイト(FDB)は0のままとされている。そこで、ステップS59、S60の処理はスキップされ、ステップS61の処理が直ちに実行される。

【0084】以上のように、この実施例によれば、一度も書き込み処理が行われていないディスクに対しては、余分な書き込み(異常検出バイト(FDB)の書き込み)が行われないことになる。

【0085】この図8の実施例は、次のような効果を有する。すなわち、電源オン時やディスク挿入時には、ボリュームディスクリプタ(VD)は必ず読み込む必要があるものであり、異常検出バイト(FDB)の検査による効率低下は発生しない。

【0086】また、ディスクへの書き込みが発生した場合、いずれにしてもボリュームスペースビットマップ(VSB)やボリュームディスクリプタ(VD)に変更が加えられるので、異常検出バイト(FDB)の記録再生を行わないとしても、ボリュームディスクリプタ(VD)、ボリュームスペースビットマップ(VSB)およびマネージメントテーブル(MT)の書き込み処理が行われる。従って、ディスクに対して書き込み処理が行われる場合には、異常検出バイト(FDB)をディスクに記録するようにしたとしても、それに起因して効率が低下するようなことはない。

【0087】すなわち、異常検出バイト(FDB)が0である場合において、データを最初に記録するときのボリュームディスクリプタ(VD)、ボリュームスペースビットマップ(VSB)およびマネージメントテーブル(MT)の書き込み(ステップS49またはS54における書き込み)のみが、異常検出バイト(FDB)を採用しない場合に較べて余分な処理となるだけである。従って、異常検出バイト(FDB)を採用したとしても、それによる効率低下はほとんど無視することができる。

【0088】次に修復プログラム実行の処理(図4のステップS13または図8のステップS43の処理)の詳細について、さらに説明する。

【0089】MDデータにおいては、データ入力部7よりCPU5に対してミラーモードの設定を指令すると、CPU5は、通常、図17に示すように、クラスタ83乃至クラスタ114に形成するボリューム管理領域(VMA)を、図9に示すように、2倍に拡大し、クラスタ83乃至クラスタ146にボリューム管理領域(VMA)を形成する。そして、ボリューム管理領域(VMA)に、実質的に同一の情報を二重書きするようにする。

【0090】すなわち、図9に示すように、各クラスタを前半と後半に区分し、前半と後半に同一の情報を記録するようにする。例えば、クラスタ83の前半には、ボリュームディスクリプタ(VD)、ボリュームスペースビットマップ(VSB)およびマネージメントテーブル(MT)と、ディレクトリレコードブロック(DRB)6乃至DRB15が記録される。そして、クラスタ83の後半にも、前半と同一のデータが記録される。クラスタ84においても、その前半と後半には、それぞれディレクトリ、レコードブロックDRB16乃至DRB31のデータが二重に書き込まれる。クラスタ85乃至クラスタ146においても同様である。

【0091】このようなミラーモードが設定された場

合、CPU5は、図4のステップS13または図8のステップS43の修復プログラム実行処理において、例えば図10に示す処理を実行する。

【0092】この処理は、上述したように、図4のステップS12、または図8のステップS42において、異常検出フラグFDBが0×42であると判定されたとき行われるものである。いずれかの位置に異常（破壊されたデータ）が記録されていることになる。

【0093】そこで、最初にステップS71において、CPU5は、クラスタ（例えばクラスタ83）の前半のデータが読み取り可能であるか否かを判定する。図11に示すように、クラスタ83の前半と後半のうち、後半のデータが破壊されていたとすれば、前半のデータ（更新された新しいデータ）は読み取ることが可能である。この場合においては、ステップS72に進み、クラスタ83の前半のボリュームディスクリプタ（VD）が読み取られる。

【0094】そして、その読み取ったボリュームディスクリプタ（VD）中の異常検出バイト（FDB）が0であるか否かをステップS73で判定する。異常検出バイト（FDB）が0である場合、他のクラスタの書き換えが既に完了しているため、データは正常であると判定することができる。そこで、この場合においては、図8のステップS46（または図4のステップS16）に進み、正常である場合と同様の処理を実行する。

【0095】これに対して、クラスタ83の前半のボリュームディスクリプタ（VD）中の異常検出バイト（FDB）が0ではない（0×42である）と判定された場合、他のクラスタは、書き換えられた可能性がある。そこで、この場合においてはステップS74に進み、修復プログラム処理が実行される。

【0096】一方、ステップS71において、クラスタ83の前半のデータが読み取り可能ではないと判定された場合、ステップS75に進み、クラスタ83の後半のデータが読み取り可能であるか否かが判定される。例えば、図12に示すように、クラスタ83の前半のデータが破壊されていたとしても、その後半のデータ（更新されていない古いデータ）が破壊されていない場合がある。この場合、ステップS76に進み、クラスタ83の後半のデータが読み取られる。そして、ステップS77において、その読み取ったボリュームディスクリプタ（VD）中の異常検出バイト（FDB）が0であるか否かを判定する。

【0097】古いボリュームディスクリプタ（VD）の異常検出バイト（FDB）が0である場合、他のクラスタは、まだ書き換えられていないので正常である。そこで、図8のステップS46（または図4のステップS16）に進み、それ以降の処理を実行する。

【0098】ステップS77において、異常検出フラグFDBが0ではない（0×42である）と判定された場

合、他のクラスタは書き換えられた可能性があるため、この場合はステップS74に進み、修復プログラム処理を実行する。

【0099】図13に示すように、クラスタ83の中央部が破壊された場合、クラスタ83の前半と後半の両方が読み取り不可となる。この場合は、ステップS71とステップS75において、NOの判定が行われるため、ステップS74に進み、修復プログラム処理が実行される。

【0100】また、クラスタ83以外のクラスタ84乃至クラスタ146のうちのいずれかのクラスタの書き込み中にシステムがダウンした場合、図14に示すように、そのクラスタnの前半（図14（a））、後半（図14（b））または中央（図14（c））において、データが破壊されている。この場合においても、クラスタ83における場合と同様に処理することができる。

【0101】なお、データが破壊されているか否かは、再生時、データに同期した同期信号がなくなること、あるいはエラー訂正符号によりエラーが検出されることなどから検出することができる。

【0102】上記実施例においては、異常検出バイトをデータ記録媒体1に記録するようにしたが、データ記録媒体1が、データ記録再生装置から取り外しできないものである場合においては、不揮発性メモリ4に書き込むようにすることもできる。

【0103】また、異常検出バイト（FDB）は、1バイト全てを用いる必要はなく、単に異常の有無を判定するだけでよいのなら、1ビットでもよい。逆に異常の有無だけでなく、ファイル管理情報の整合性、ファイル中のデータの整合性など、より多様な整合性判定を行う場合においては、より多くのビットを使用するようにすることもできる。さらにこの場合においては、異常検出符号の他に、変更を加えようとするファイル名などを記録するようにすることもできる。

【0104】以上の実施例においては、データ記録媒体1をMDデータとしたが、この発明は、MDデータ以外の光磁気ディスクはもとより、その他の磁気ディスク、ICメモ리카ードなどのデータ記録媒体にも適用することが可能である。

【0105】

【発明の効果】以上の如く請求項1に記載のデータ記録再生装置および請求項8に記載のデータ記録方法によれば、第1の識別符号の記録の後に、データを記録媒体に記録し、データを記録した後に、第1の識別符号を第2の識別符号に書き換えるようにしたので、識別符号からデータ管理情報の整合性を瞬時に判定することができる。

【0106】その結果、データ記録再生装置の起動時やデータ記録媒体の交換時に、毎回整合性判定を行ったとしても、作業効率が低下することが抑制される。そし

て、整合性が取れていない状態において、データ記録媒体に対して、データの記録再生処理が行われないうにすることができるので、操作性および信頼性を飛躍的に向上させることが可能となる。

【0107】また、請求項9に記載のデータ記録媒体によれば、データを記録する前の状態、またはデータを記録した後の状態であることを表す識別符号を記録するようにしたので、データ記録再生装置から取り外し可能なデータ記録媒体においても、記録されているデータの整合が取れているか否かを、迅速かつ確実に、判定することが可能となる。

【図面の簡単な説明】

【図1】本発明のデータ記録再生装置の構成例を示すブロック図である。

【図2】図1のデータ記録媒体1のフォーマットを説明する図である。

【図3】図1の実施例の第1の処理例を示すフローチャートである。

【図4】図1の実施例の第2の処理例を示すフローチャートである。

【図5】図4のステップS19の記録処理の詳細を説明するフローチャートである。

【図6】図4のステップS20の再生処理の詳細を説明するフローチャートである。

【図7】図4のステップS21の消去処理の詳細を説明するフローチャートである。

【図8】図1の実施例の第3の処理例を示すフローチャートである。

【図9】ミラーモード時におけるフォーマットを説明す

る図である。

【図10】ミラーモード時における図8のステップS43における処理の詳細を説明するフローチャートである。

【図11】クラスタ83の後半が破壊された状態を示す図である。

【図12】クラスタ83の前半が破壊された状態を示す図である。

【図13】クラスタ83の中央部が破壊された状態を示す図である。

【図14】クラスタ83以外のクラスタが破壊された状態を示す図である。

【図15】従来のMDデータのフォーマットを示す図である。

【図16】ボリューム管理領域の構成を説明する図である。

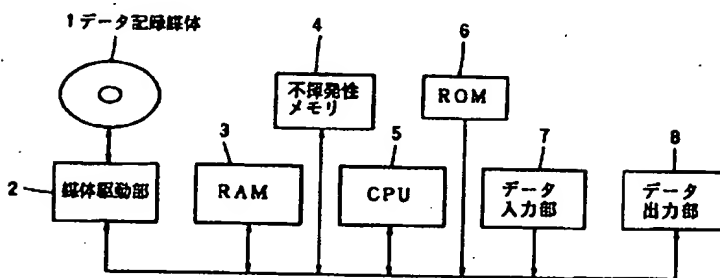
【図17】クラスタ83乃至クラスタ114のデータの配置を説明する図である。

【図18】従来のデータ記録再生装置の処理例を示すフローチャートである。

【符号の説明】

- 1 データ記録媒体
- 2 媒体駆動部
- 3 RAM
- 4 不揮発性メモリ
- 5 CPU
- 6 ROM
- 7 データ入力部
- 8 データ出力部

【図1】

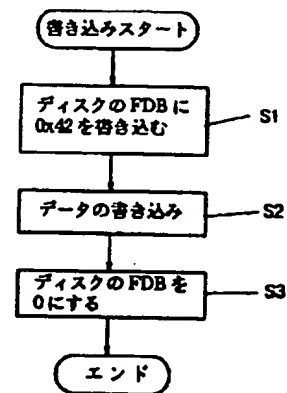


【図11】

クラスタ83の後半が破壊された場合

クラスタ 83	VD	VSB	MT	DRB6-DRB15	新 XXXXXXXX	ID
------------	----	-----	----	------------	------------	----

【図3】

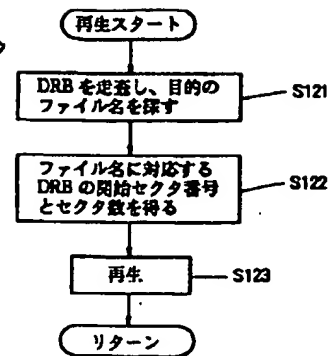


【図12】

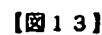
クラスタ83の前半が破壊された場合

クラスタ 83	新 XXXXXXXX	ID	VD	VSB	MT	DRB6-DRB15
------------	------------	----	----	-----	----	------------

【圖 6】

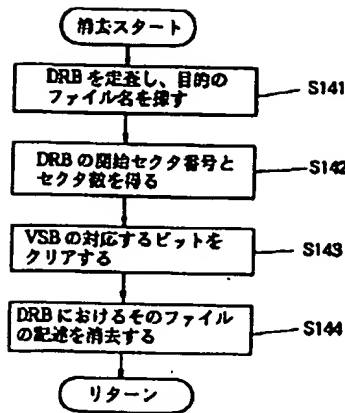


【圖5】

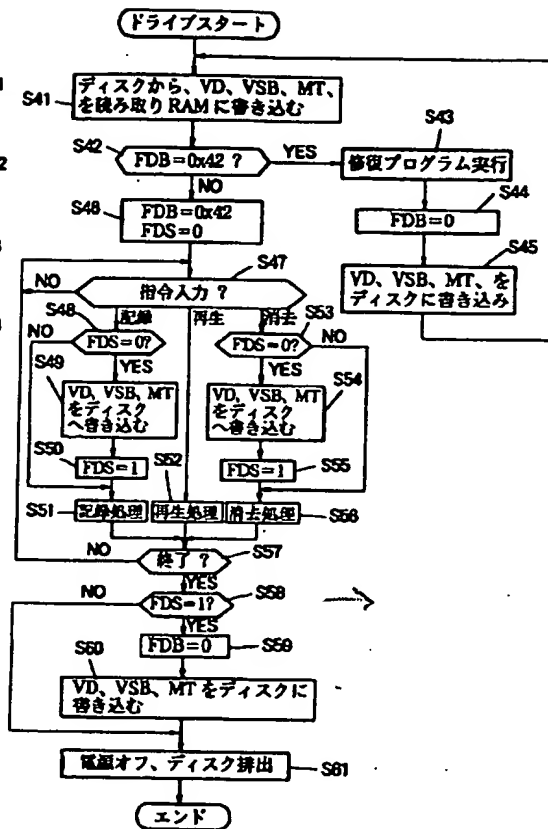


クラス 83 VD VSB MT DRB 8 XXX XXXX IB MT DRB8-DRB15

【図 7】

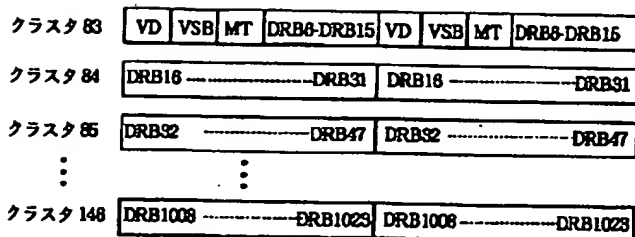


【図 8】

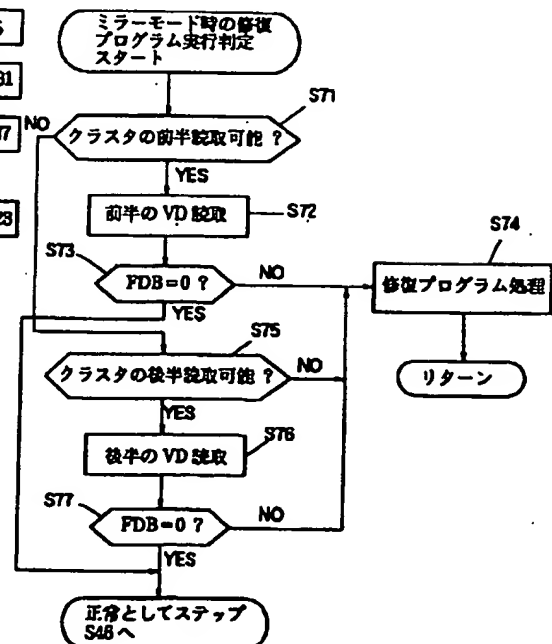
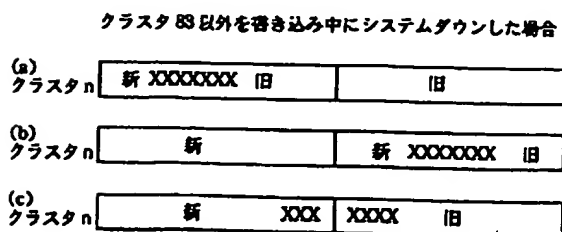


【図 9】

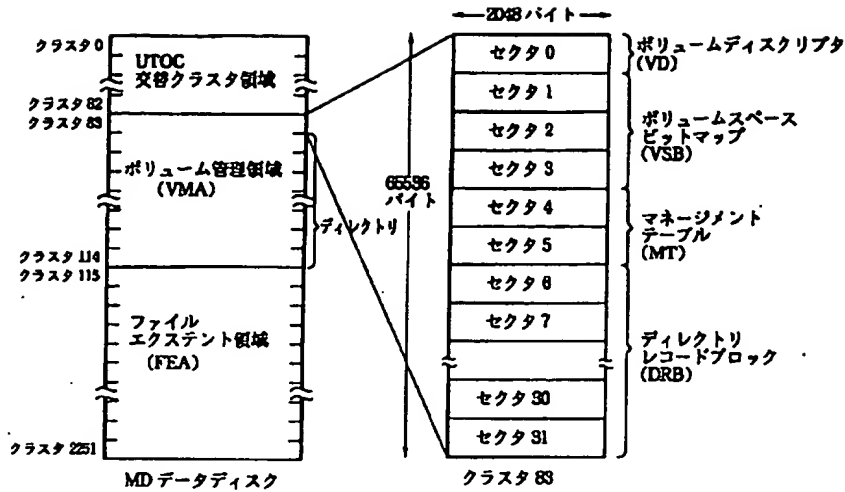
【図 10】



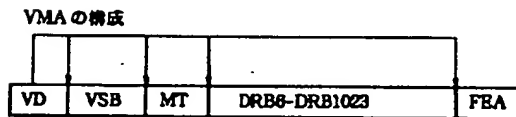
【図 14】



【図15】

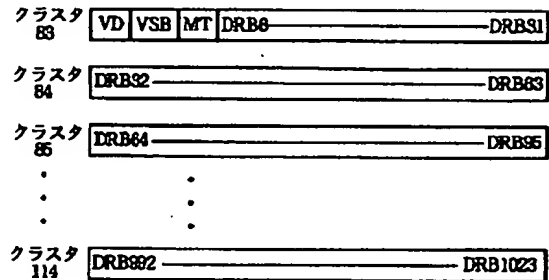


【図16】

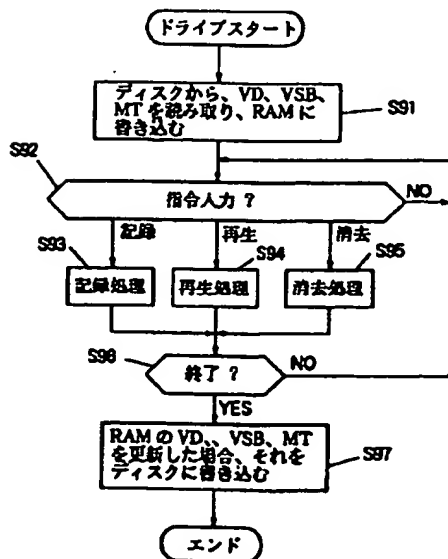


VD (Volume Descriptor) ポリューム情報の管理
 VSB (Volume Space Bitmap) FEA 中の空き領域管理
 MT (Management Table) 空き DRB の管理
 DRB (Directory Record Block) ファイル/ディレクトリの管理

【図17】



【図18】



【公報種別】特許法第17条の2の規定による補正の掲載

【部門区分】第6部門第4区分

【発行日】平成14年1月11日(2002. 1. 11)

【公開番号】特開平9-167447

【公開日】平成9年6月24日(1997. 6. 24)

【年通号数】公開特許公報9-1675

【出願番号】特願平7-330059

【国際特許分類第7版】

G11B 20/12 102

【F I】

G11B 20/12 102

【手続補正書】

【提出日】平成13年7月4日(2001. 7. 4)

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】発明の名称

【補正方法】変更

【補正内容】

【発明の名称】データ記録装置および方法並びにデータ記録媒体

【手続補正2】

【補正対象書類名】明細書

【補正対象項目名】特許請求の範囲

【補正方法】変更

【補正内容】

【特許請求の範囲】

【請求項1】 記録媒体にデータ記録する前に、前記データを記録する前の状態であることを表す第1の識別符号を記録する第1の記録手段と、
前記第1の識別符号の記録の後に、前記データを前記記録媒体に記録する第2の記録手段と、
前記第2の記録手段による前記データの記録の後に、前記第1の識別符号を、前記データを記録した後の状態であることを表す第2の識別符号に書き換える第3の記録手段とを備えることを特徴とするデータ記録装置。

【請求項2】 前記第1の記録手段は、前記データ記録装置の使用が開始されたとき、前記第1の識別符号を記録することを特徴とする請求項1に記載のデータ記録装置。

【請求項3】 前記記録媒体から、前記第1または第2の識別符号を再生する再生手段と、
前記再生手段により再生された前記第1または第2の識別符号に対応して、前記記録媒体の修復処理を行う修復手段とをさらに備えることを特徴とする請求項1に記載のデータ記録装置。

【請求項4】 前記第3の記録手段は、前記データ記録装置の動作終了が指令されたとき、前記第2の識別符号を記録することを特徴とする請求項1に記載のデータ記

録装置。

【請求項5】 前記第1の記録手段と第3の記録手段は、前記第1の識別符号と前記第2の識別符号を、前記記録媒体に記録することを特徴とする請求項1に記載のデータ記録装置。

【請求項6】 前記記録媒体に対して前記データを記録することを表す第3の識別符号を記録する第4の記録手段をさらに備え、
前記第1の記録手段は、前記記録媒体に対して前記データを実際に記録するとき、前記第1の識別符号を記録することを特徴とする請求項1に記載のデータ記録装置。

【請求項7】 前記第1の記録手段と第3の記録手段は、前記第1の識別符号と第2の識別符号を、複数回、前記記録媒体に記録することを特徴とする請求項1に記載のデータ記録装置。

【請求項8】 記録媒体にデータ記録する前に、前記データを記録する前の状態であることを表す第1の識別符号を記録する第1の記録ステップと、
前記第1の識別符号の記録の後に、前記データを前記記録媒体に記録する第2の記録ステップと、
前記データの記録の後に、前記第1の識別符号を、前記データを記録した後の状態であることを表す第2の識別符号に書き換える第3の記録ステップとを含むことを特徴とするデータ記録方法。

【請求項9】 前記第1の記録ステップは、前記データ記録方法の使用が開始されたとき、前記第1の識別符号を記録することを特徴とする請求項8に記載のデータ記録方法。

【請求項10】 前記記録媒体から、前記第1または第2の識別符号を再生する再生ステップと、
前記再生ステップにより再生された前記第1または第2の識別符号に対応して、前記記録媒体の修復処理を行う修復ステップとをさらに含むことを特徴とする請求項8に記載のデータ記録方法。

【請求項11】 前記第3の記録ステップは、前記データ記録方法の動作終了が指令されたとき、前記第2の識

別符号を記録することを特徴とする請求項8に記載のデータ記録方法。

【請求項12】 前記第1の記録ステップと第3の記録ステップは、前記第1の識別符号と前記第2の識別符号を、前記記録媒体に記録することを特徴とする請求項8に記載のデータ記録方法。

【請求項13】 前記記録媒体に対して前記データを記録することを表す第3の識別符号を記録する第4の記録ステップをさらに含み、前記第1の記録ステップは、前記記録媒体に対して前記データを実際に記録するとき、前記第1の識別符号を記録することを特徴とする請求項8に記載のデータ記録方法。

【請求項14】 前記第1の記録ステップと第3の記録ステップは、前記第1の識別符号と第2の識別符号を、複数回、前記記録媒体に記録することを特徴とする請求項8に記載のデータ記録方法。

【請求項15】 データを記録するデータ記録媒体において、前記データとともに、前記データを記録する前の状態、または前記データを記録した後の状態であることを表す識別符号が記録されていることを特徴とするデータ記録媒体。

【手続補正3】

【補正対象書類名】明細書

【補正対象項目名】0001

【補正方法】変更

【補正内容】

【0001】

【発明の属する技術分野】本発明はデータ記録装置および方法並びにデータ記録媒体に関し、特に、データ記録媒体中に記録されているデータを管理する管理情報が正しいか否かを迅速に識別することができるようにした、データ記録装置および方法並びにデータ記録媒体に関する。

【手続補正4】

【補正対象書類名】明細書

【補正対象項目名】0031

【補正方法】変更

【補正内容】

【0031】

【課題を解決するための手段】請求項1に記載のデータ記録装置は、記録媒体にデータ記録する前に、データを記録する前の状態であることを表す第1の識別符号を記録する第1の記録手段と、第1の識別符号の記録の後に、データを記録媒体に記録する第2の記録手段と、第2の記録手段によるデータの記録の後に、第1の識別符号を、データを記録した後の状態であることを表す第2の識別符号に書き換える第3の記録手段とを備えることを特徴とする。

【手続補正5】

【補正対象書類名】明細書

【補正対象項目名】0032

【補正方法】変更

【補正内容】

【0032】請求項8に記載のデータ記録方法は、記録媒体にデータ記録する前に、データを記録する前の状態であることを表す第1の識別符号を記録する第1の記録ステップと、第1の識別符号の記録の後に、データを記録媒体に記録する第2の記録ステップと、データの記録の後に、第1の識別符号を、データを記録した後の状態であることを表す第2の識別符号に書き換える第3の記録ステップとを含むことを特徴とする。

【手続補正6】

【補正対象書類名】明細書

【補正対象項目名】0033

【補正方法】変更

【補正内容】

【0033】請求項15に記載のデータ記録媒体は、データとともに、データを記録する前の状態、またはデータを記録した後の状態であることを表す識別符号が記録されていることを特徴とする。

【手続補正7】

【補正対象書類名】明細書

【補正対象項目名】0036

【補正方法】変更

【補正内容】

【0036】請求項15に記載のデータ記録媒体においては、データとともに、データを記録する前の状態、またはデータを記録した後の状態であることを表す識別符号が記録される。

【手続補正8】

【補正対象書類名】明細書

【補正対象項目名】0105

【補正方法】変更

【補正内容】

【0105】

【発明の効果】以上の如く請求項1に記載のデータ記録装置および請求項8に記載のデータ記録方法によれば、第1の識別符号の記録の後に、データを記録媒体に記録し、データを記録した後に、第1の識別符号を第2の識別符号に書き換えるようにしたので、識別符号からデータ管理情報の整合性を瞬時に判定することができる。

【手続補正9】

【補正対象書類名】明細書

【補正対象項目名】0107

【補正方法】変更

【補正内容】

【0107】また、請求項15に記載のデータ記録媒体によれば、データを記録する前の状態、またはデータを記録した後の状態であることを表す識別符号を記録する

ようにしたので、データ記録再生装置から取り外し可能なデータ記録媒体においても、記録されているデータの

整合が取れているか否かを、迅速かつ確実に、判定することが可能となる。

* NOTICES *

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] Especially this invention relates to a data-logging medium at the data-logging regenerative apparatus and approach list from which the management information which manages the data currently recorded on the data-logging regenerative apparatus and the approach list into the data-logging medium about the data-logging medium enabled it to discriminate quickly whether it is the right.

[0002]

[Description of the Prior Art] Recently, various data have come to be recorded and used for data-logging media, such as a magnetic disk, a magneto-optic disk, and IC memory card, with the spread of computers.

[0003] Drawing 15 expresses the example of a configuration of the file system of MD (mini disc) (trademark) data which are a kind of a magneto-optic disk. As shown in this drawing, MD data disk is classified into 2252 clusters, and 83 fields of a cluster 0 thru/or a cluster 82 are made into UTOC and a shift cluster field, and let the cluster 83 thru/or the cluster 114 be volume-management fields (Volume Management Area (VMA)). Moreover, let the cluster 115 thru/or the cluster 2251 be the file extent fields (File Extent Area (FEA)) which record data.

[0004] The cluster 3 in UTOC and a shift cluster field thru/or 5 are made into a UTOC (User Table of Content) field, and let a cluster 50 thru/or 82 be shift cluster fields. Other clusters are considered as reserve.

[0005] In the cluster 83 of the head of a volume-management field (VMA) A volume descriptor (Volume Descriptor (VD)) is arranged at the sector of the head. Into the 3 following sectors Into 2 sectors which a volume space bit map (Volume Space Bitmap (VSB)) is arranged, and continue A management table (MangementTable (MT)) is arranged and the directory record block (Directory Record Block (DRB)) is arranged at 26 sectors which continue further.

[0006] One cluster is constituted by 32 sectors and 1 sector is made into 2048 bytes. Therefore, one cluster becomes 65536 bytes (64 K bytes).

[0007] The volume descriptor (VD) expresses the location of the volume space bit map (VSB) currently recorded on this disk, a management table (MT), a directory record block (DRB), and a file extent field (FEA), as shown in drawing 16 .

[0008] The volume space bit map (VSB) has bit map data which manage the free area of a file extent field (FEA). The management table (MT) has managed the opening of a directory record block (DRB). Moreover, let a directory record block (DRB) be the field which performs management of a file and a directory. One DRB is described by one sector.

[0009] Drawing 17 expresses FOMATTO of a volume-management field (VMA) typically. As shown in this drawing, into the sector of the beginning of a cluster 83, a volume space bit map (VSB) is arranged at the 3 following sectors, and the management table (MT) is arranged for the volume descriptor (VD) at the 2 following sectors, respectively. And the directory record block (DRB) 6 thru/or DRB1023 are

arranged at each sector after it of a cluster 83, and each sector of a cluster 84 thru/or a cluster 114.

[0010] Drawing 18 expresses the example of processing in the conventional MD data record regenerative apparatus.

[0011] First, in step S91, from a disk (MD data), a volume descriptor (VD), a volume space bit map (VSB), and a management table (MT) are read, and it is written in RAM (not shown) of a data-logging regenerative apparatus. Next, in step S92, it stands by until one command of record, playback, and elimination is inputted.

[0012] When record, playback, or the command of elimination is inputted, it progresses to steps S93, S94, or S95, respectively, and record processing corresponding to a command, regeneration, or elimination processing is performed. That is, data are recorded, reproduced or eliminated to the file extent field (FEA) of a disk.

[0013] Next, it is judged whether in step S96, the command of power-source OFF or the command which discharges a disk was inputted, and termination of operation was directed. When termination is not directed, return and processing after it are repeated and performed to step S92.

[0014] When it judges that it was ordered by termination in step S96, it progresses to step S97, and when the contents of the volume descriptor (VD) memorized by RAM, a volume space bit map (VSB), and the management table (MT) are updated, it is written in a disk.

[0015] Thus, it is made as [access / the file of the arbitration currently recorded on the file extent field (FEA) with reference to the contents of the volume-management field (VMA) / by updating the contents of the volume-management field (VMA) corresponding to it, when data are recorded on a file extent field (FEA) or the data currently recorded are eliminated / freely].

[0016] However, if it collects at the end and the contents are written in a disk when memorizing all the contents of the volume-management field (VMA) to RAM in this way and terminating actuation, capacity of RAM must be enlarged, after ordering it termination it not only becoming cost quantity, but, time amount until actuation is actually ended will become long, and operability will get worse.

[0017] Then, recording the contents on a disk, whenever it changes the contents of the volume-management field (VMA) is also considered. However, when it is made such, processing speed in use will fall.

[0018] In order to solve this, a volume descriptor (VD), a volume space bit map (VSB), A management table (MT) is stationed permanently at RAM. And a directory record block (DRB) When making it make what has high access frequency memorize on RAM and accessing a new directory record block (DRB) The directory record block (DRB) which became old is written in on a disk, and the Least Recently Used (LRU) algorithm which incorporates a new directory record block (DRB) to RAM may be used.

[0019] However, if it is made such, the information which is not written in a disk yet will remain on RAM, and as a result of the information that operating frequency is higher residing in RAM permanently, moreover, possibility of not being written in a disk will become high.

[0020] Consequently, if the condition (a system down is carried out) that a data-logging regenerative apparatus stops occurs in the condition that the operations sequence of the file system mentioned above is not completed by failure of equipment, interruption of service, a failure, etc., a system error will occur.

[0021] For example, since the volume space bit map (VSB) is not updated yet to the timing with which the predetermined file A was registered into the directory record block (DRB) 32, and the contents data was also written to be to the file extent field (FEA) supposing a system down occurs, it will be considered that the field where the file A of a file extent field (FEA) is recorded at the time of future accesses is a free area. Consequently, if it is going to record the file B new next, File B will be overwritten by the field to which File A is recorded, and the situation where File A will be eliminated will occur.

[0022] Moreover, for example, in the directory record block (DRB) 6 of a parent directory, Subdirectory C is registered as a directory record block (DRB) 64, and if a system down is carried out in the condition that the directory record block (DRB) 64 is not updated yet, Subdirectory C will not exist in fact.

Consequently, it will become a system error if Subdirectory C is read.

[0023]

[Problem(s) to be Solved by the Invention] Thus, if abnormalities are in the file management information on a disk, it will become difficult to hold a file (data) correctly. So, when it inspects whether abnormalities are in the file management information on a disk and abnormalities are discovered, he is trying to restore this.

[0024] Inspection of whether abnormalities are in the file management information on a disk is conducted by inspecting the following items, for example.

[0025] (1) Does the subdirectory registered into each directory actually exist or not?

[0026] (2) Has the volume space bit map (VSB) become under use to the record section of each file registered into each directory or not?

[0027] (3) In addition, isn't there any self-conflict of file management information or not?

[0028] The time amount for several minutes is needed for these inspection. This inspection can be conducted, when a data-logging regenerative apparatus is equipped with a disk there when the power source of for example, a data-logging regenerative apparatus is turned on or. However, when it is made such, before actually beginning to use a data-logging regenerative apparatus, time amount will be taken too much, and operability will be spoiled.

[0029] Then, when abnormalities are accepted in actuation of equipment, it can inspect. However, when it is made such, after important data are destroyed, abnormalities will be discovered and dependability will be spoiled.

[0030] This invention is made in view of such a situation, and the data control information on a record medium enables it to detect whether it is the right quickly and certainly.

[0031]

[Means for Solving the Problem] The 1st record means which records the 1st identification code showing being in the condition before recording data before carrying out data logging of the data-logging regenerative apparatus according to claim 1 to a record medium, It is characterized by having the 3rd record means rewritten to the 2nd identification code showing being in the condition after recording data for the 1st identification code after record of the data based on the 2nd record means which records data on a record medium, and the 2nd record means after record of the 1st identification code.

[0032] The data-logging approach according to claim 8 is characterized by rewriting to the 2nd identification code showing being in the condition after recording the 1st identification code showing being in the condition before recording data before carrying out data logging to a record medium, recording data on a record medium and recording data for the 1st identification code after record of data after record of the 1st identification code.

[0033] A data-logging medium according to claim 9 is characterized by recording the identification code showing being in the condition before recording data with data, or the condition after recording data.

[0034] In a data-logging regenerative apparatus according to claim 1 The 1st identification code showing being in the condition before carrying out data logging to a record medium and the 1st record means records data is recorded. The 2nd record means records data on a record medium after record of the 1st identification code, and rewrites to the 2nd identification code showing the 3rd record means being in the condition after recording data for the 1st identification code after record of the data based on the 2nd record means.

[0035] In the data-logging approach according to claim 8, it rewrites to the 2nd identification code showing being in the condition after recording the 1st identification code showing being in the condition before recording data before carrying out data logging to a record medium, recording data on a record medium and recording data for the 1st identification code after record of data after record of the 1st identification code.

[0036] In a data-logging medium according to claim 9, the identification code showing being in the condition before recording data with data, or the condition after recording data is recorded.

[0037]

[Embodiment of the Invention] Drawing 1 is the block diagram showing the example of a configuration of the data-logging regenerative apparatus of this invention. The data-logging medium 1 is a record medium which consists of a magnetic disk, a magneto-optic disk, an IC memory card, etc., and data are recorded. The medium mechanical component 2 is constituted by the disk drive equipment which consists of a strange demodulator circuit which records data to the data-logging medium 1, and reproduces this, modulates the spindle motor turning around this, the optical pickup which irradiates a laser beam to a magneto-optic disk, the servo circuit which performs various kinds of servo actuation, and a record signal when the data-logging medium 1 is [for example,] a magneto-optic disk, and restores to a regenerative signal.

[0038] RAM3 functions also as a work area which performs function as a buffer in the case of carrying out record playback of the data to the data-logging medium 1, and actuation of various kinds [CPU /5] upwards, and memorizes required data, a required program, etc. suitably. Nonvolatile memory 4 memorizes data with the need of memorizing even after turning off the power source of a data-logging regenerative apparatus. CPU5 performs various kinds of processings according to the program memorized by ROM6.

[0039] The data input section 7 inputs various kinds of data, and is constituted by a keyboard, a digital interface, the A/D converter, etc. The data output section 8 is a part which outputs data, for example, is constituted by a digital interface, a CRT display, the D/A converter, etc.

[0040] In the following examples, the data-logging medium 1 presupposes that it is an MD data.

[0041] Drawing 2 expresses the format of the MD data as a data-logging medium 1. Although this format is fundamentally [as FOMA@TTO of the conventional MD data explained with reference to drawing 15] the same, in this example, a malfunction detection cutting tool (Fault Detect Byte (FDB)) is stationed at the intact cutting tool of a volume descriptor (VD) stationed at the cluster 83 of a volume-management field (VMA). Other configurations are the same as that of the case in drawing 15.

[0042] if, as for this malfunction detection cutting tool (FDB), that value becomes zero -- normal -- if it becomes 0x42, it will be defined as abnormalities. In addition, 0x express a hexadecimal here.

[0043] This malfunction detection cutting tool (FDB) is recorded as shown in the flow chart of drawing 3. That is, when a user operates the data input section 7 and orders it record of data now, CPU5 controls the medium mechanical component 2, and makes the malfunction detection cutting tool (FDB) in the volume descriptor (VD) of the MD data as a data-logging medium 1 write in 0x42 (the 1st identification code) in step S1 (1st record means).

[0044] Next, it progresses to step S2 (2nd record means), and CPU5 makes RAM3 memorize temporarily the record data inputted from the data input section 7. And the data memorized by RAM3 are read, and this is supplied to the data-logging medium 1, and is made to record on a file extent field (FEA) through the medium mechanical component 2. The management information of a volume-management field (VMA) is also updated with record of this data.

[0045] And it progresses to step S3 (3rd record means), and CPU5 rewrites the malfunction detection cutting tool (FDB) in the disk (MD data) which wrote in 0x42 at step S1 to 0 (the 2nd identification code).

[0046] Thus, since 0x42 is written in a malfunction detection cutting tool (FDB) and he is trying to rewrite that value to 0 after write-in completion of data according to this example before writing data in an MD data, when a system down occurs in the middle of rewriting processing of data, the malfunction detection cutting tool (FDB) of an MD data becomes as [0x42].

[0047] On the other hand, the contents of the disk are normal even if a system down occurs the writing front of data, and after the completion of rewriting. And a malfunction detection cutting tool's (FDB) value is 0 at this time.

[0048] Then, when a data-logging regenerative apparatus is equipped with a disk, or when the power source of a data-logging regenerative apparatus is turned on, the malfunction detection cutting tool (FDB) of an MD data is read, and when the value is 0x42, it is made to continue subsequent processing as it is, if the value is 0, and to perform the restoration program which restores the contents of the disk.

[0049] However, according to the example of drawing 3, processing (processing of steps S1 and S3)

which rewrites a malfunction detection cutting tool (FDB) whenever it rewrites the data on a disk must be performed, and processing speed falls. As shown in the flow chart of drawing 4, it can process there. [0050] CPU5 reads a volume descriptor (VD), a volume space bit map (VSB), and a management table (MT) in the data-logging medium 1 (MD data), and makes RAM3 memorize it in step S11 (playback means) in processing of drawing 4 first. Next, in step S12, the malfunction detection cutting tool (FDB) in the volume descriptor (VD) read now is read, and it judges whether the value is 0x42.

[0051] When a system down occurs in the middle of record actuation of data, also in this example, the malfunction detection cutting tool (FDB) is 0x42, so that it may mention later. Then, it judges with the contents of data not being normal in this case, and progresses to step S13 (restoration means), and a restoration program is performed. Thereby, the abnormalities in the data-logging medium 1 are restored. That is, restoration is performed so that it may be made as [exist / the subdirectory registered into the directory / actually], and it may become under use of a volume space bit map (VSB) to the record section of each file registered into each directory, in addition there may be no self-conflict of management information.

[0052] Since such restoration was performed, next it progresses to step S14, and the value of the malfunction detection cutting tool (FDB) in the volume descriptor on RAM3 (VD) is rewritten to 0. And the volume descriptor (VD), volume space bit map (VSB), and management table (MT) which are memorized by RAM3 are made to write in the data-logging medium 1 in step S15 (the contents of the MD data are updated). By this, 0 will be recorded on the volume descriptor (VD) of an MD data as a malfunction detection cutting tool's (FDB) value.

[0053] Then, return and a malfunction detection cutting tool's (FDB) value written in the RAM3 when the volume descriptor (VD), the volume space bit map (VSB), and the management table (MT) were again read in the MD data and having been written in RAM3 are step S11 with 0. Then, in this case, in step S12, the judgment of NO is performed and it progresses to step S16.

[0054] A malfunction detection cutting tool's (FDB) value memorized by RAM3 is made to be rewritten from 0 to 0x42 in step S16. And in step S17, the volume descriptor (VD), volume space bit map (VSB), and management table (MT) which are memorized by RAM3 are read, and it is written in an MD data.

[0055] Processing of this drawing 4 is started, when a data-logging regenerative apparatus is equipped with the data-logging medium 1, or when the data-logging regenerative apparatus is already equipped with the data-logging medium 1 and it turns on the power source of a data-logging regenerative apparatus. That is, when starting use of a data-logging regenerative apparatus, it performs.

[0056] Therefore, before starting the substantial use to the data-logging medium 1, the data-logging medium 1 is made into an always normal condition. Moreover, in this condition, the malfunction detection cutting tool (FDB) of the data-logging medium 1 has, when a system down occurs in the middle after that, and that value is set as 0x42.

[0057] Next, it progresses to step S21 and CPU5 performs elimination processing, when it progresses to step S20 and regeneration is performed, when it progresses to step S19 and record processing is performed, when it stands by until the command of predetermined actuation was inputted from the data input section 7 in step S18, and ordered in record, and ordered in playback, and ordered in elimination.

[0058] Drawing 5 expresses the detail of record processing. First, in step S111, CPU5 scans the volume space bit map (VSB) memorized by RAM3, records data above, and searches a required number of empty sectors. And it progresses to step S112 and data are recorded on the empty sector searched with step S111.

[0059] In addition, in the case of an MD data, record is performed per cluster. Therefore, when data are already recorded on other sectors of the cluster to which it is vacant and a sector belongs which should newly record data, the data of the sector are read, and RAM3 memorizes. And the record data inputted from the data input section 7 are written in into the empty sector of the cluster of RAM3. And after the data for one cluster are collected, the data for the one cluster are read from RAM3, and are written in the cluster of the file extent field (FEA) where an MD data corresponds through the medium mechanical component 2.

[0060] Next, it progresses to step S113, and it is vacant and CPU5 sets as used the bit corresponding to a

sector which recorded data now [of the volume space bit map (VSB) memorized by RAM3].

[0061] Next, it progresses to step S114, the directory record block (DRB) corresponding to the sector which recorded data is now read from the data-logging medium 1, and it memorizes to RAM3. And in step S115, the file name of the file constituted with the data recorded on the directory record block (DRB) read now at step S112, the number of the first sector which recorded data, the number of the sectors which recorded data, etc. are recorded, and this is further recorded on an MD data.

[0062] Drawing 6 expresses the detail of regeneration. In regeneration, in step S121, CPU5 reads the directory record block (DRB) currently recorded on the data-logging medium 1 to RAM3, and scans it, and the file name which was inputted from the data input section 7 and which should be reproduced is searched. When the directory record block (DRB) of the target file name is searched, it progresses to step S122 and the initiation sector number and the number of sectors of the file are read in the directory record block (DRB). And in step S123, CPU5 reproduces a number of sectors read too from the initiation sector number read at step S122.

[0063] After this playback data is memorized by RAM3, it is read again and outputted to the data output section 8.

[0064] Drawing 7 expresses the detail of elimination processing. In elimination processing, in step S141, the directory record block (DRB) of the data-logging medium 1 is read to RAM, and is scanned, and the directory record block (DRB) which has the file name made applicable to elimination is searched. When the directory record block (DRB) which has the target file name is searched, it progresses to step S142 and the initiation sector number and the number of sectors of the directory record block (DRB) are read.

[0065] Next, it progresses to step S143 and CPU5 clears the bit to which it corresponds on the volume space bit map (VSB) of a number of sectors which constitute the file from the sector of the initiation sector number read at step S142. This volume space bit map (VSB) is memorized by RAM3. And it progresses to step S144 and the description of the file in the directory record block (DRB) in the data-logging medium 1 is eliminated.

[0066] It judges whether it returned to drawing 4, and after record processing, regeneration, or elimination processing was performed [in / as mentioned above / step S19 thru/or S21], it progressed to step S22 and was ordered in termination of processing. If discharge of the data-logging medium 1, the command which turns off the power source of a data-logging regenerative apparatus are inputted from the data input section 7 and it is not ordered in termination, return and processing after it are repeated and performed to step S18.

[0067] When it judges that it was ordered by termination of processing, it progresses to step S23 and CPU5 rewrites the malfunction detection cutting tool (FDB) of RAM3 to 0. At the time of processing initiation, this malfunction detection cutting tool (FDB) is step S16, and is rewritten by 0x42. And it progresses to step S24 and the volume descriptor (VD), volume space bit map (VSB), and management table (MT) which are memorized by RAM3 are written in the data-logging medium 1. Furthermore, in step S25, corresponding to a termination command, a power source is turned off or processing which discharges a disk is performed.

[0068] As mentioned above, since processing which rewrites a malfunction detection cutting tool (FDB) by a unit of 1 time, respectively to the substantial step S18 thru/or busy condition front of S22 and the back is only performed according to the example of drawing 4, the fall of processing speed is controlled.

[0069] However, not only when writing data in a disk in the case of the example shown in drawing 4 but when only reproducing, write-in processing of a malfunction detection cutting tool (FDB) will be performed. Then, when only regeneration which does not perform record actuation of data to the data-logging medium 1 is performed, it is also possible for it to be made not to update the malfunction detection cutting tool (FDB) of the data-logging medium 1, either. Drawing 8 expresses the example in this case.

[0070] In drawing 8, first, in step S41, a volume descriptor (VD), a volume space bit map (VSB), and a management table (MT) are read in the data-logging medium 1, and RAM3 memorizes. And in step S42, it is judged now whether the malfunction detection cutting tool (FDB) in the read volume

descriptor (VD) is 0x42. When a malfunction detection cutting tool (FDB) is 0x42, it progresses to step S43, and a restoration program is performed. And after restoration processing is performed, in step S44, the malfunction detection cutting tool (FDB) of RAM3 is rewritten by 0, and the volume descriptor (VD), volume space bit map (VSB), and management table (MT) which are memorized by RAM3 are written in the data-logging medium 1 in step S45. The above processings are step S11 in drawing 4 thru/or processing of S15 and the same processing.

[0071] In step S42, when judged with the malfunction detection cutting tool (FDB) who read to RAM3 not being 0x42, it progresses to step S46 (4th record means) (when judged with it being 0), and CPU5 rewrites the value of the malfunction detection cutting tool (FDB) of RAM3 from 0 to 0x42. Moreover, the inspection cutting tool (FDS) (the 3rd identification code) memorized by RAM3 (or nonvolatile memory 4) is set as 0. This inspection cutting tool (FDS) functions as an identification code showing whether data were written in to the data-logging medium 1. Here, a value 0 is initialized as that by which data are not written in the data-logging medium 1.

[0072] Next, it progresses to step S47, and it stands by until a predetermined command is inputted from the data input section 7.

[0073] When ordered in record of data, it progresses to step S48 and CPU5 judges whether the inspection cutting tool (FDS) memorized by RAM3 (again nonvolatile memory 4) is 0. When data were not once written in to the data-logging medium 1 yet, as it mentioned above, 0 is initialized by the inspection cutting tool (FDS) at step S46. In this case, it progresses to step S49 and the volume descriptor (VD), volume space bit map (VSB), and management table (MT) which are memorized by RAM3 are written in the data-logging medium 1. Therefore, the malfunction detection cutting tool (FDB) of the data-logging medium 1 is set to 0x42.

[0074] Next, it progresses to step S50 and CPU5 rewrites to 1 an inspection cutting tool's (FDS) value memorized by RAM3. Thereby, it is set up that record processing was performed to the data-logging medium 1 (carried out at continuing step S51). And it progresses to step S51, and is inputted from the data input section 7, and the data memorized to RAM3 are made to record on the file extent field (FEA) of the data-logging medium 1. The detail of the record processing in this step S51 is the same as that of the case where it is shown in drawing 5.

[0075] When judged with an inspection cutting tool (FDS) not being 0 (it being 1), it means that write-in processing of data was performed once or more to the data-logging medium 1 in step S48. In this case, since the malfunction detection cutting tool (FDB) of the data-logging medium 1 is already rewritten by 0x42, processing of steps S49 and S50 is skipped, and progresses to step S51, and record processing is already started immediately.

[0076] On the other hand, also in this case, when ordered in elimination of data in step S47, since the writing of the data-logging medium 1 is performed, the same processing as the case in record is performed. That is, in this case, if it is judged whether the inspection cutting tool (FDS) memorized by RAM3 is 0 and it is 0 in step S53, it will progress to step S54, and the volume descriptor (VD), volume space bit map (VSB), and management table (MT) which are memorized by RAM3 are written in the data-logging medium 1. And in step S55, the inspection cutting tool (FDS) memorized by RAM3 is set as 1. Then, it progresses to step S56 and elimination processing is performed. The detail of this elimination processing is the same as that of the case where it is shown in drawing 7.

[0077] In step S53, when judged with an inspection cutting tool (FDS) not being 0 (it being 1), write-in processing of data will already have been performed to the data-logging medium 1 once or more. Then, the malfunction detection cutting tool (FDB) is already rewritten by the value of 0x42 in this case. for this reason, the step S -- processing of 54 and 55 is skipped and elimination processing of step S56 is performed immediately.

[0078] Furthermore, in step S47, when a reproductive command is inputted, it progresses to step S52 and regeneration is performed. The detail of this regeneration is the same as that of the case where it is shown in drawing 6.

[0079] That is, when ordered in playback, unlike the case where it is ordered in record or elimination, processing which updates the malfunction detection cutting tool (FDB) of the data-logging medium 1

will be performed.

[0080] Next it is steps S51, S52, or S56, it progresses to step S57, and if not judged and ordered [whether it was ordered in termination of processing, and], return and processing after it are repeatedly performed by step S47.

[0081] When it judges that it was ordered by termination of processing, it progresses to step S58 and it is judged whether an inspection cutting tool (FDS) is 1. As mentioned above, when data are recorded even once to the data-logging medium 1, in steps S50 or S55, 1 is assigned to an inspection cutting tool (FDS), and the malfunction detection cutting tool (FDB) in the data-logging medium 1 is rewritten by the value of 0x42 in steps S49 or S54. Then, in step S59, the malfunction detection cutting tool (FDB) in the volume descriptor (VD) of RAM3 is rewritten to 0. And in step S60, processing which writes the volume descriptor (VD), volume space bit map (VSB), and management table (MT) which are memorized by RAM3 in the data-logging medium 1 is performed.

[0082] Furthermore, it progresses to step S61 and processing which discharges power-source OFF or a disk is performed corresponding to the command of termination.

[0083] In step S58, when judged with an inspection cutting tool (FDS) not being 1 (it being 0), even if regeneration is performed, also at once, write-in processing (record or elimination processing) of data will be performed to the data-logging medium 1. Therefore, the malfunction detection cutting tool (FDB) of the data-logging medium 1 is considered as as [0]. Then, processing of steps S59 and S60 is skipped, and processing of step S61 is performed immediately.

[0084] As mentioned above, according to this example, also at once to the disk with which write-in processing is not performed, excessive writing (a malfunction detection cutting tool's (FDB) writing) will be performed.

[0085] The example of this drawing 8 has the following effectiveness. That is, it is necessary to surely read a volume descriptor (VD) at the time of power-source ON and disk insertion, and it does not generate the degradation by inspection of a malfunction detection cutting tool (FDB) at it.

[0086] Moreover, though a malfunction detection cutting tool's (FDB) record rebirth is not performed since modification is added anyway to a volume space bit map (VSB) or a volume descriptor (VD) when the writing to a disk occurs, write-in processing of a volume descriptor (VD), a volume space bit map (VSB), and a management table (MT) is performed. Effectiveness seems therefore, to originate in it and not to fall, though the malfunction detection cutting tool (FDB) was recorded on the disk when write-in processing was performed to a disk.

[0087] That is, when a malfunction detection cutting tool (FDB) is 0, only the writing (writing in steps S49 or S54) of the volume descriptor (VD) when recording data first, a volume space bit map (VSB), and a management table (MT) only serves as excessive processing compared with the case where a malfunction detection cutting tool (FDB) is not employed. Therefore, even if it employs a malfunction detection cutting tool (FDB), most degradation by it can be disregarded.

[0088] Next, the detail of processing (processing of step S13 of drawing 4 or step S43 of drawing 8) of restoration program execution is explained further.

[0089] In an MD data, if it is ordered a setup in mirror mode from the data input section 7 to CPU5, as shown in drawing 17, CPU5 will expand the volume-management field (VMA) formed in a cluster 83 thru/or a cluster 114 twice, as shown in drawing 9, and will usually form a volume-management field (VMA) in a cluster 83 thru/or a cluster 146. And it is made to carry out duplex writing of the same information to a volume-management field (VMA) substantially.

[0090] That is, as shown in drawing 9, each cluster is classified into the first half and the second half, and the same information is recorded on the first half and the second half. For example, a volume descriptor (VD), a volume space bit map (VSB) and a management table (MT), and the directory record block (DRB) 6 thru/or DRB15 is recorded in the first half of a cluster 83. And the same data as the first half are recorded also in the second half of a cluster 83. Also in a cluster 84, a directory, a record block DRB16, or the data of DRB31 is written in a duplex at the first half and second half, respectively. Also in a cluster 85 thru/or a cluster 146, it is the same.

[0091] When such mirror mode is set up, CPU5 performs processing shown in drawing 10 in restoration

program execution processing of step S13 of drawing 4 , or step S43 of drawing 8 .

[0092] Since this processing is performed when it is judged with the malfunction detection flag FDB being 0x42 in step S12 of drawing 4 , or step S42 of drawing 8 , as mentioned above, abnormalities (destroyed data) will be recorded on one of locations.

[0093] Then, in step S71, CPU5 judges first whether the data in the first half of a cluster (for example, cluster 83) can be read. If the data of the second half were destroyed while in the first half and the second half of a cluster 83 as shown in drawing 11 , the data (updated new data) of the first half can be read. In this case, it progresses to step S72 and the volume descriptor in the first half of a cluster 83 (VD) is read.

[0094] And the malfunction detection cutting tool (FDB) in the read volume descriptor (VD) judges whether it is 0 at step S73. Since rewriting of other clusters is already completed when a malfunction detection cutting tool (FDB) is 0, it can judge with data being normal. Then, in this case, it progresses to step S46 (or step S16 of drawing 4) of drawing 8 , and the same processing as the case of being normal is performed.

[0095] On the other hand, other clusters may have been rewritten when judged with the malfunction detection cutting tool (FDB) in the volume descriptor in the first half of a cluster 83 (VD) not being 0 (it being 0x42). Then, it progresses to step S74 in this case, and restoration program manipulation is performed.

[0096] On the other hand, when it judges that the data in the first half of a cluster 83 is not [reading] possible in step S71, it progresses to step S75 and it is judged whether the data in the second half of a cluster 83 can be read. For example, though the data in the first half of a cluster 83 are destroyed as shown in drawing 12 , the data (old data which are not updated) of the second half may not be destroyed. In this case, it progresses to step S76 and the data in the second half of a cluster 83 are read. And in step S77, it judges whether the malfunction detection cutting tool (FDB) in the read volume descriptor (VD) is 0.

[0097] When the malfunction detection cutting tool (FDB) of an old volume descriptor (VD) is 0, since other clusters are not rewritten yet, they are normal. Then, it progresses to step S46 (or step S16 of drawing 4) of drawing 8 , and processing after it is performed.

[0098] In step S77, since other clusters may have been rewritten when judged with the malfunction detection flag FDB not being 0 (it being 0x42), it progresses to step S74 in this case, and restoration program manipulation is performed.

[0099] As shown in drawing 13 , when the center section of the cluster 83 is destroyed, both the first half and the second half of a cluster 83 read, and it becomes improper. In this case, in step S71 and step S75, since the judgment of NO is performed, it progresses to step S74 and restoration program manipulation is performed.

[0100] Moreover, when a system is downed during the writing of clusters 84 other than cluster 83 thru/or the cluster of either of the clusters 146, as shown in drawing 14 , data are destroyed in the first half of the cluster n (drawing 14 (a)), the second half (drawing 14 (b)), or a center (drawing 14 (c)). Also in this case, it can process like the case in a cluster 83.

[0101] In addition, it is detectable at the time of playback whether data are destroyed or not from that the synchronizing signal which synchronized with data is lost, an error being detected by the error correction sign, etc.

[0102] In the above-mentioned example, although the malfunction detection cutting tool was recorded on the data-logging medium 1, when the data-logging medium 1 is what cannot be removed from a data-logging regenerative apparatus, it can write in nonvolatile memory 4.

[0103] Moreover, if a malfunction detection cutting tool (FDB) does not need to use 1 byte of all and should just only judge the existence of abnormalities, 1 bit is satisfactory for him. Conversely, more bits can be used when performing more various adjustment judgments, such as the adjustment of not only the existence of abnormalities but file management information, and the adjustment of the data in a file. The file name which is furthermore going to add modification other than a malfunction detection sign in this case can be recorded.

[0104] In the above example, although the data-logging medium 1 was made into the MD data, magneto-optic disks other than an MD data can be applied from the first also to data-logging media, such as other magnetic disks and IC memory card, by this invention.

[0105]

[Effect of the Invention] Since the 1st identification code was rewritten to the 2nd identification code after recording data on the record medium and recording data after record of the 1st identification code like the above according to a data-logging regenerative apparatus according to claim 1 and the data-logging approach according to claim 8, the adjustment of data control information can be judged from an identification code in an instant.

[0106] Consequently, even if it performs an adjustment judging each time at the time of starting of a data-logging regenerative apparatus and exchange of a data-logging medium, it is controlled that working efficiency falls. And in the condition that adjustment cannot be taken, since it can avoid performing record regeneration of data to a data-logging medium, it becomes possible to raise operability and dependability by leaps and bounds.

[0107] Moreover, since the identification code showing being in the condition before recording data, or the condition after recording data was recorded according to the data-logging medium according to claim 9, also in a dismountable data-logging medium, it becomes [whether adjustment of the data currently recorded can be taken, and] possible quickly and certainly from a data-logging regenerative apparatus to judge.

[Translation done.]

* NOTICES *

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] The 1st record means which records the 1st identification code showing being in the condition before recording said data before carrying out data logging to a record medium, The 2nd record means which records said data on said record medium after record of said 1st identification code, The data-logging regenerative apparatus characterized by having the 3rd record means rewritten to the 2nd identification code showing being in the condition after recording said data for said 1st identification code after record of said data based on said 2nd record means.

[Claim 2] Said 1st record means is a data-logging regenerative apparatus according to claim 1 characterized by recording said 1st identification code when use of said data-logging regenerative apparatus is started.

[Claim 3] The data-logging regenerative apparatus according to claim 1 characterized by having further a playback means to reproduce said 1st or 2nd identification code, and a restoration means to perform restoration processing of said record medium corresponding to said 1st or 2nd identification code reproduced by said playback means, from said record medium.

[Claim 4] Said 3rd record means is a data-logging regenerative apparatus according to claim 1 characterized by recording said 2nd identification code when ordered in termination of said data-logging regenerative apparatus of operation.

[Claim 5] Said 1st record means and the 3rd record means are a data-logging regenerative apparatus according to claim 1 characterized by recording said the 1st identification code and said 2nd identification code on said record medium.

[Claim 6] It is the data-logging regenerative apparatus according to claim 1 characterized by recording said 1st identification code when it has further the 4th record means which records the 3rd identification code showing recording said data to said record medium and said 1st record means actually records said data to said record medium.

[Claim 7] Said 1st record means and the 3rd record means are a data-logging regenerative apparatus according to claim 1 characterized by recording said the 1st identification code and 2nd identification code on multiple times and said record medium.

[Claim 8] The data-logging approach characterized by rewriting to the 2nd identification code showing being in the condition after recording the 1st identification code showing being in the condition before recording said data before carrying out data logging to a record medium, recording said data on said record medium and recording said data for said 1st identification code after record of said data after record of said 1st identification code.

[Claim 9] The data-logging medium characterized by recording the identification code which expresses that it is in the condition before recording said data with said data, or the condition after recording said data in the data-logging medium which records data.

[Translation done.]